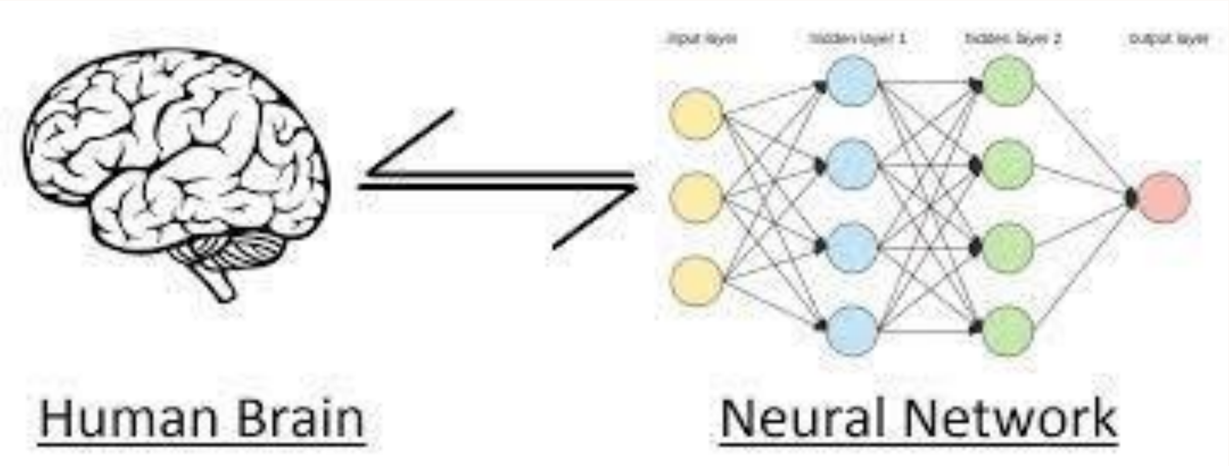
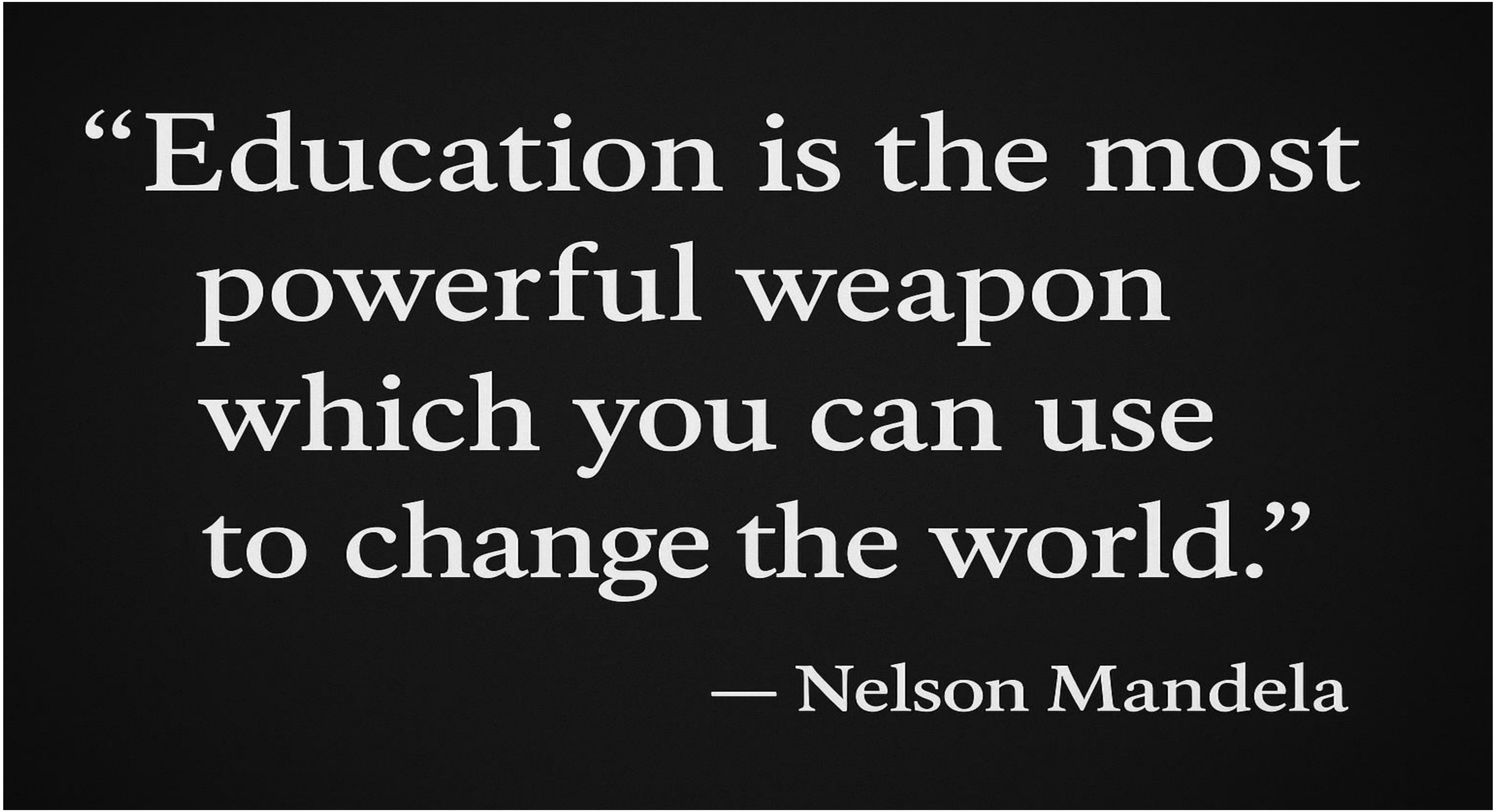


Artificial Neural Network and Deep Learning

Course Instructor

Dr.Amr Ibrahim

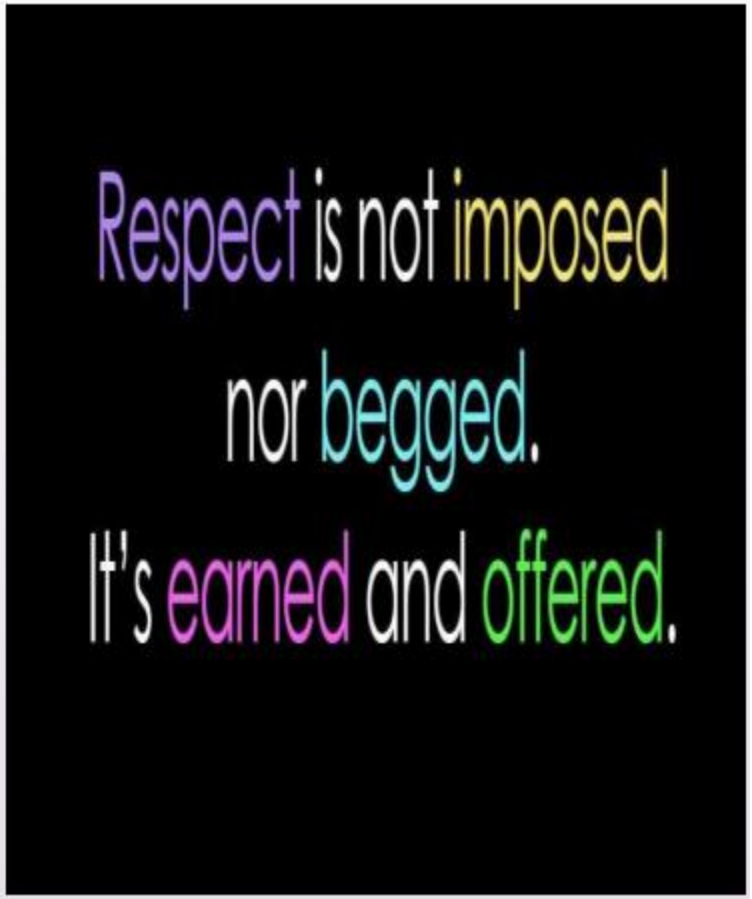


> “Education is the most powerful weapon which you can use to change the world.”
>
> — Nelson Mandela

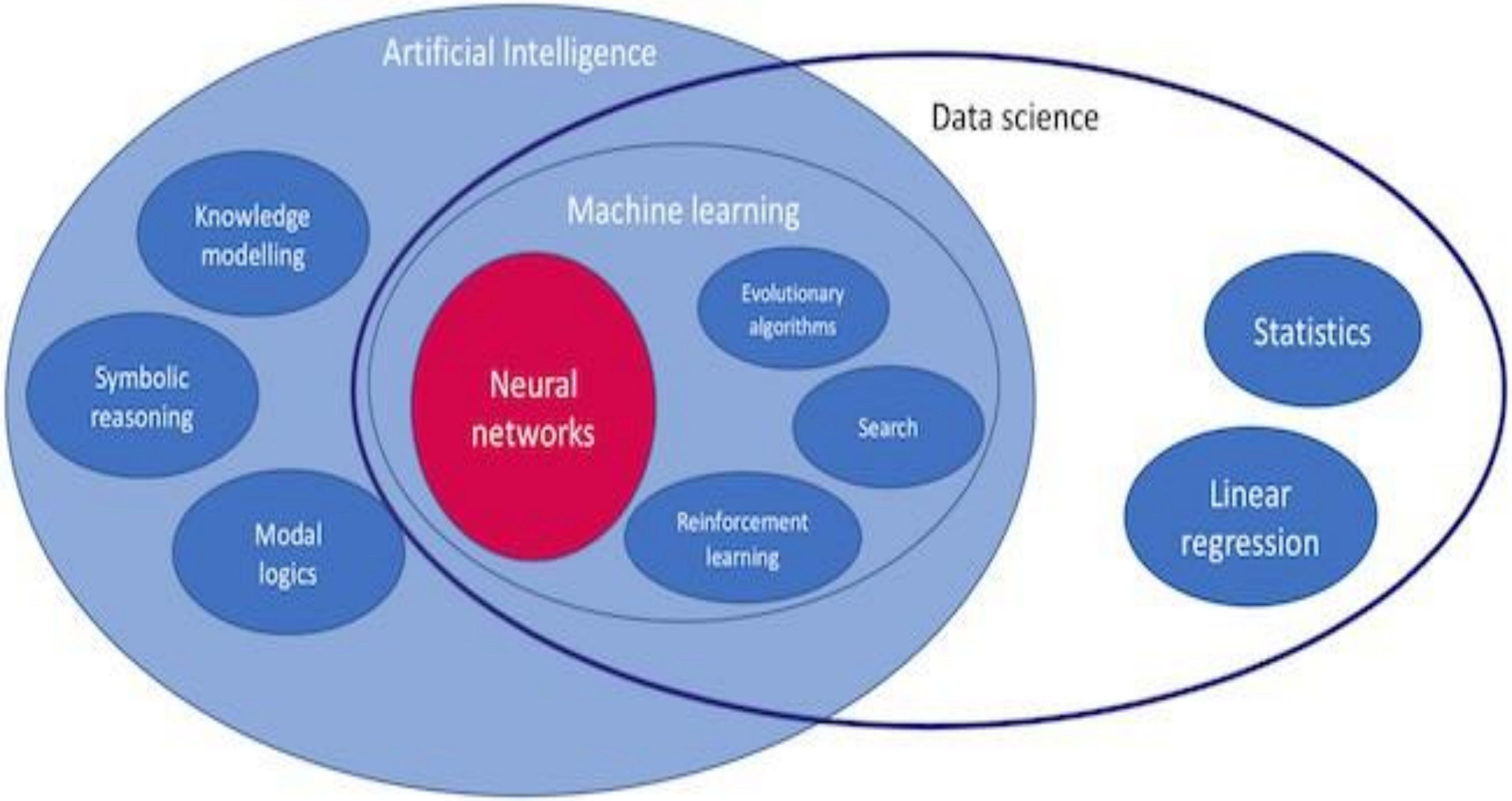
Expected Code of Conduct

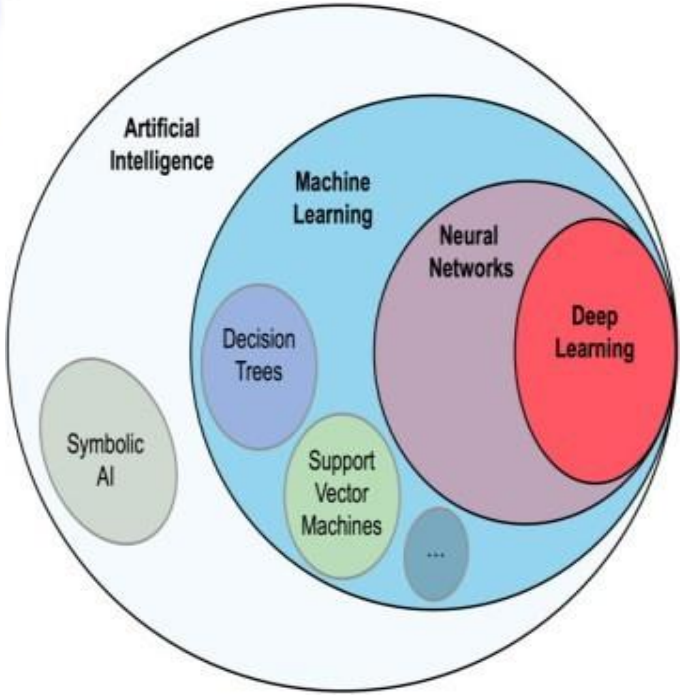
Thankyou For Being
On Time!

NO
FOOD OR
BEVERAGES
ALLOWED
IN THIS AREA


Respect is not imposed nor begged.
It's earned and offered.

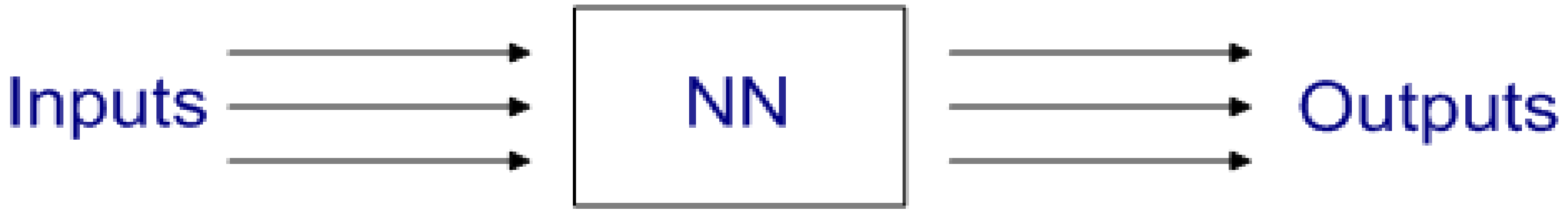
Where is NN?

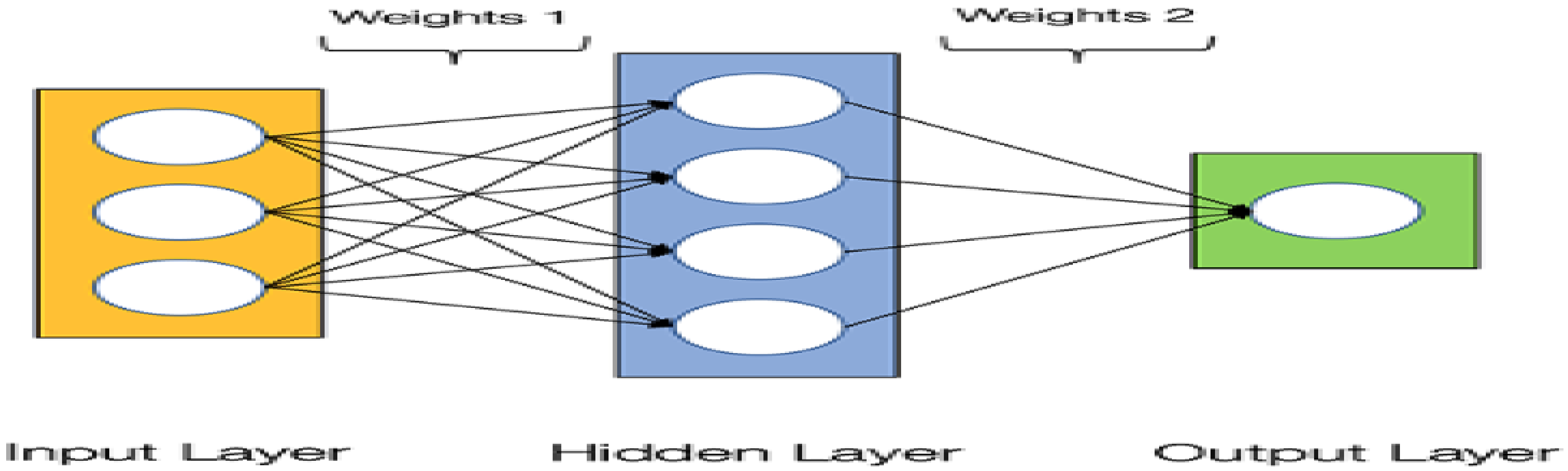

Artificial Intelligence
Data science
Machine learning
Knowledge modelling
Symbolic reasoning
Modal logics
Neural networks
Evolutionary algorithms
Search
Reinforcement learning
Statistics
Linear regression


Artificial Intelligence
Machine Learning
Neural Networks
Deep Learning
Decision Trees
Symbolic AI
Support Vector Machines
...

What is a Neural Network?, cont.

- The researchers considered the neural network as a black box strategy, which is trainable.





Architecture of a 2-layer Neural Network

Classes of Neural Network Structures

- An architecture is the way in which the neurons are connected together.

- We may identify two fundamentally different classes of network architectures (Structures):

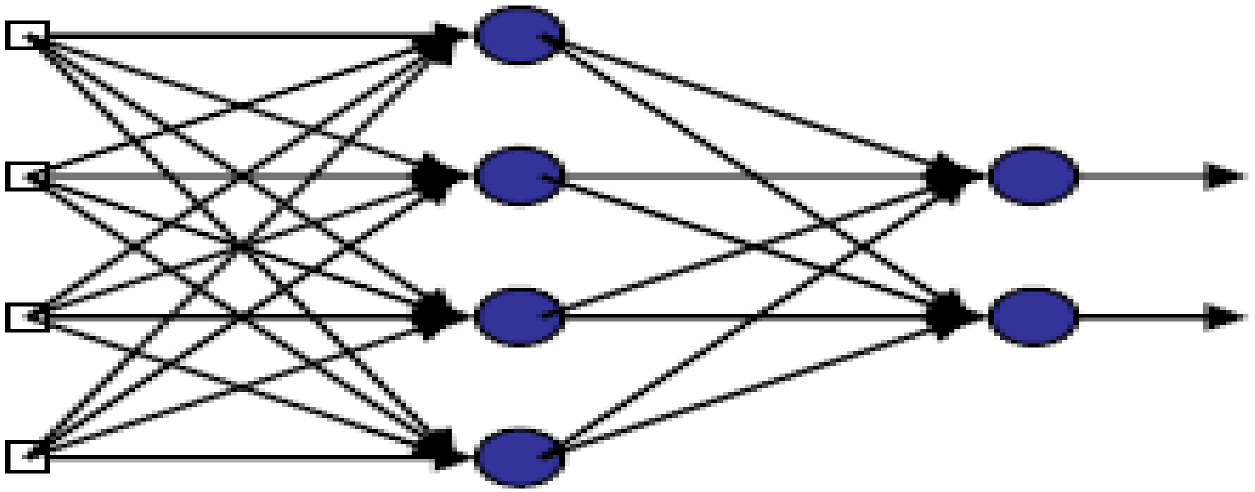
1. Feedforward Networks

2. Backforward or Recurrent Networks

1- Feedforward Networks

These are the commonest type of neural network in practical applications.

- The first layer is the input and the last layer is the output.
- If there is more than one hidden layer, we call them "deep" neural networks.

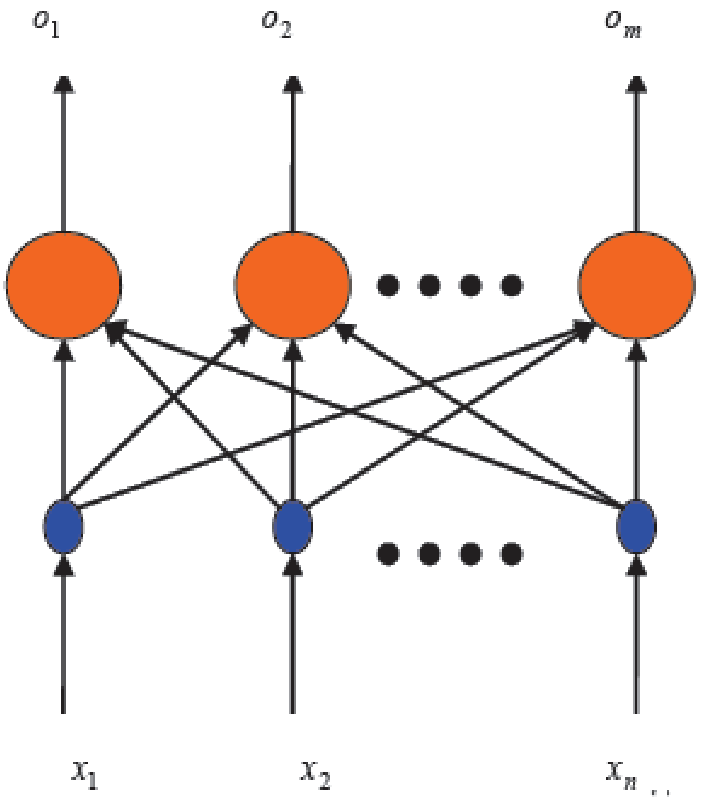


Feedforward Networks Characteristics

- ❑**Hierarchical:** the neurons are arranged in separate layers
- ❑ There is no connection between the neurons in the same layer
- ❑ This network is strictly a Feedforward, in which graphs have no loops.
- ❑The connections are unidirectional
- ❑The neurons in one layer receive inputs from the previous layer
- ❑The neurons in one layer delivers its output to the next layer.

Feedforward Networks, cont. Types

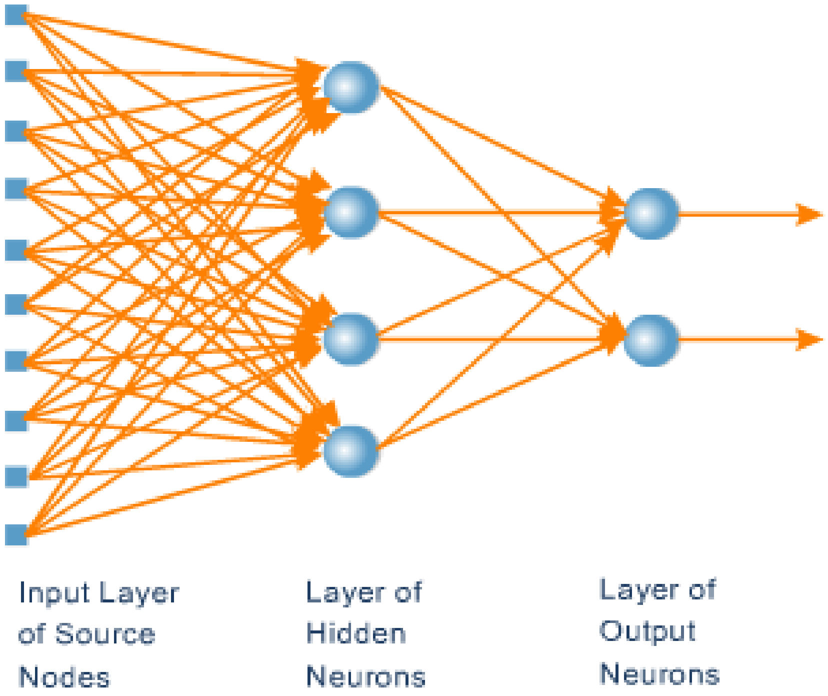
1. **Single-layer Feedforward Networks**
   - **Single-Layer Feedforward** Network is the simplest form of layered network.
   - The “signal-layer” referring to the output layer.
   - It has an **input layer** of source nodes that projects onto an **output layer** of neurons (computation nodes), but not vice versa.
   - A network is called a *single-layer network*, because **we do not count the input layer since no computation is performed there.**



Feedforward Networks, cont. Types

2. Multilayer Feedforward Networks

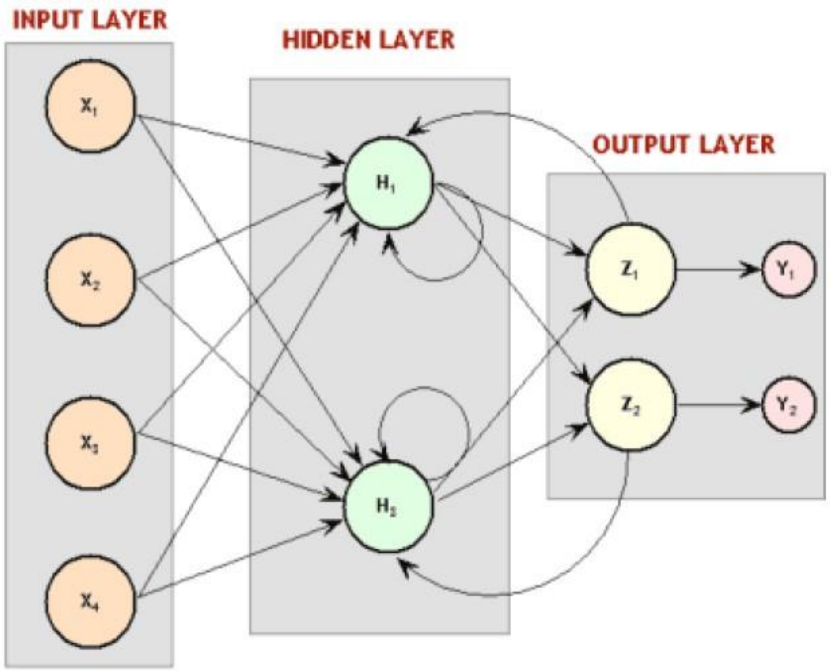
- **Contains:**
  - **Input layer (source nodes)**
  - **one or more hidden layers, whose computation nodes are called hidden neurons or hidden units.**
  - **One output layer**
- **Example**: 10-4-2 network, because it has 10 source node, 4 hidden neurons, and 2 output neurons.



➢ It is said to be ***fully connected*** in the sense that every node in each layer of the network is connected to every other node in the adjacent forward layer; otherwise, it is called ***partially connected*** if some of the weight connections are missing from the network.

2- Recurrent Networks

- It has at least one *feedback* loop.
- The network may have or not hidden neurons.
- There could be neurons with self-feedback links; that is the output of a neuron is fed back into its self as input.
- They are more biologically realistic.



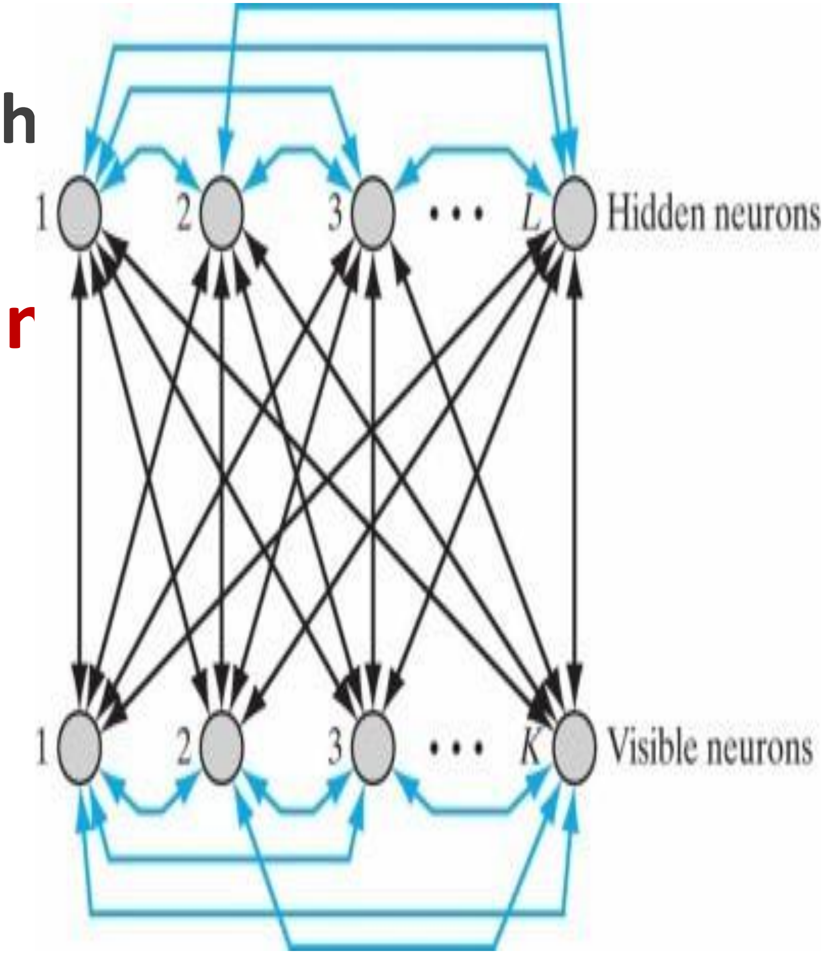
- Recurrent neural networks are **a very natural way** to model **sequential_data**. For example, it is the most appropriate for **predicting** the price of a stock.
- Feedback connection takes the output of the previous data in a series as its next input.
- They have the ability to **remember** information in their hidden state for a **long time.**

2- Recurrent Networks

Another example: Boltzmann machines

- It is a Symmetrically connected networks with hidden units.

- They are much more powerful models than
- Hopfield nets.
- They are less powerful than recurrent neural networks.
- They have a beautifully simple learning algorithm.



Learning in Neural Networks

Learning

- ❑ is a process to **store** the **information** into the network.
- ❑ defines how the system "**adapts**" to **new knowledge**.
- ❑ **On-line learning** and **off-line learning**.

Learning rules, for a connectionist system, are **algorithms** or **equations** which govern **changes** in the **weights** of the **connections in a network**.

Learning Approaches in NNs

❑ **The learning methods in Neural Networks are classified into two basic types:**

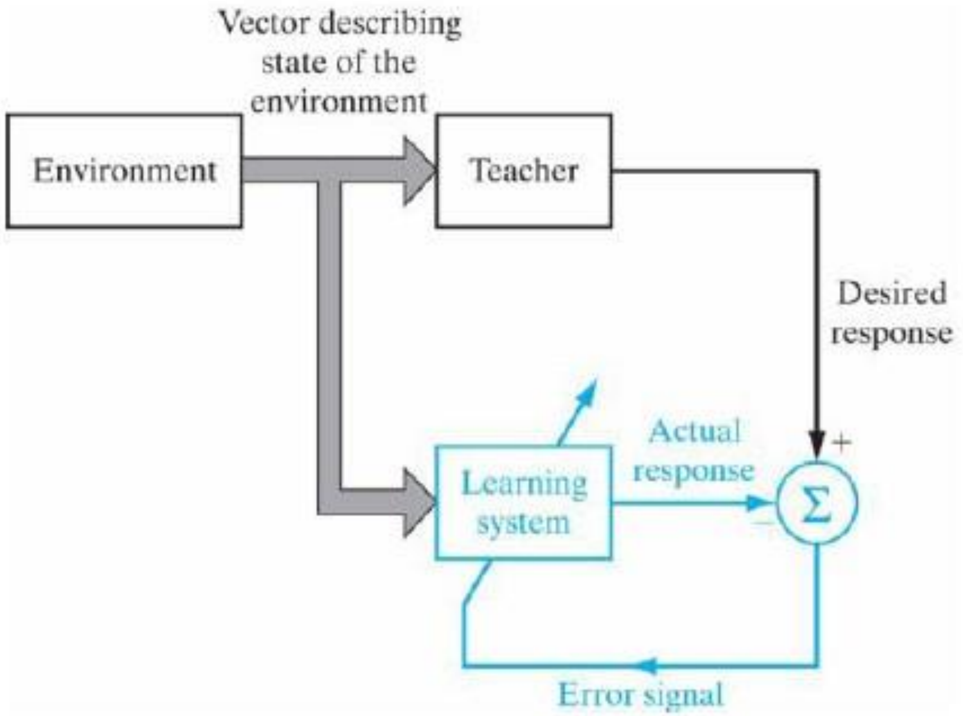
1. **Learning with a Teacher (Supervised Learning)**
2. **Learning without Teacher (Unsupervised Learning)**

❑ **These two types are classified based on:**

- ➢ **presence or absence** of teacher and
- ➢ the **information** provided for the system to learn.

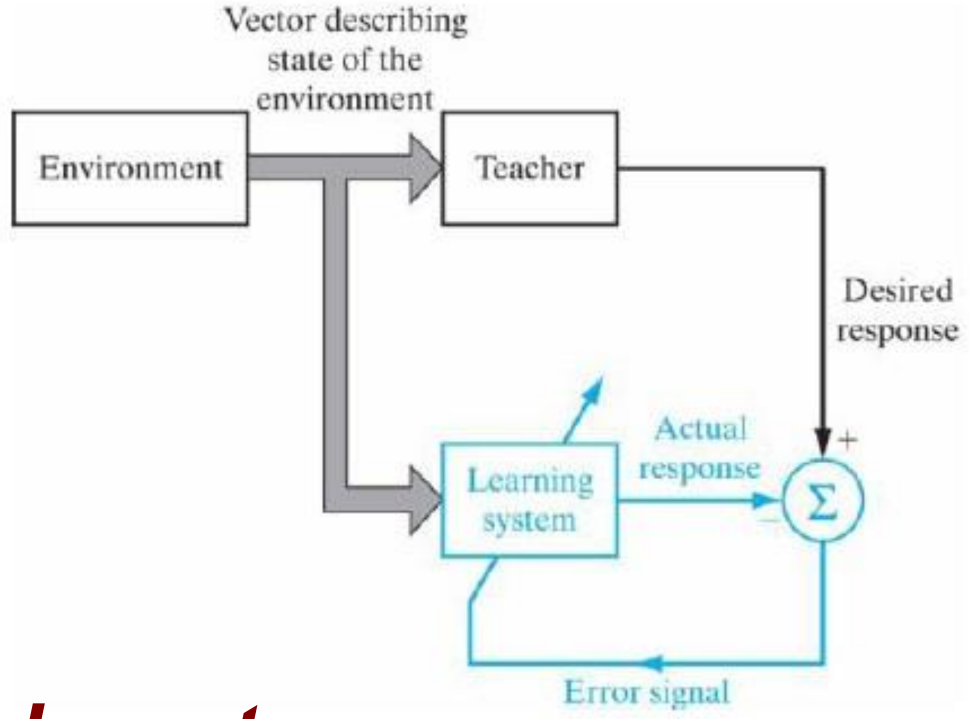
1- Learning with a teacher (Supervised Learning)

- Supervised learning is the problem of finding an input-output mapping from empirical data.
- The teacher has knowledge of the environment.
  - The teacher is able to provide the neural network with a desired response for the training vector.
  - So the NN is supplied with a sequence of labeled training patterns representing different classes.
  - Each training Labeled pattern contains input signals (features), and the desires output class, (x_i, d_i).



1- Learning with a teacher (Supervised Learning)

- The learning algorithm tries to **minimize the error** between the **desired response** t and the **actual output** y.
- In this way, the **connection** strengths of NN (**Weights**) are modified depending on
  - the **input signal** receives,
  - its **output value** (actual response)
  - and the **desired response**.



- The supervised learning process constitutes ***a closed-loop feedback system***.

1- Learning with a teacher (Supervised Learning)

Properties:

- ❑ Adaptively
  - ➢ Adapt weights to the environment
- ❑ Generalization ability

Supervised Learning algorithms:

- ❑ Perceptron learning algorithm.
- ❑ Error Correction Learning
  - ➢ Least Mean Square (LMS).
  - ➢ Back-propagation algorithm.

Tasks:

- ❑ Pattern classification (The target output is a class label)
- ❑ Object Recognition
- ❑ Regression (target output is a real number)

2- Learning without a Teacher (Unsupervised Learning)

- ❑ The teacher's response is not available
- ❑ No error signal is available
- ❑ When no teacher's response is available the NN will modify its weight based only on the input.

Unsupervised Learning is the problem of finding structure in data.

Tasks:

- Dimensionality reduction
- Clustering

2- Unsupervised Learning, cont.

Unsupervised Learning algorithms :

- **Hebbian Learning**
  - **Used for Dimensionality reduction**
  - **Similar to Principal Component Analysis (PCA)**
- **Competitive learning**
  - **Used for Clustering - The NN must identify clusters in the input data and discover classes automatically**
  - **Self-organizing features map (SOFM) are neural network model for unsupervised learning.**

2- Unsupervised Learning, cont.

Dimensionality reduction used as preprocessing step in Pattern Recognition schema.

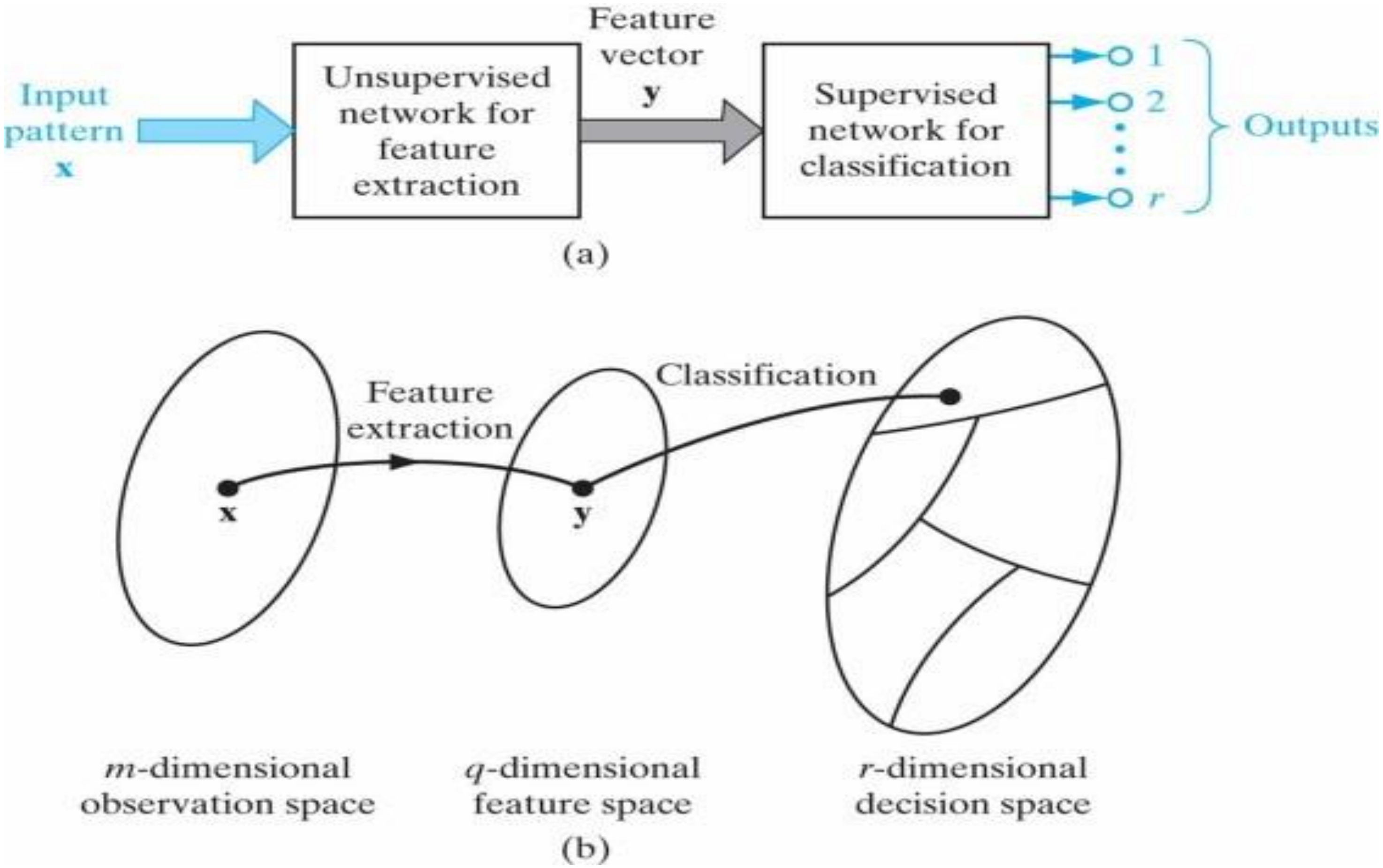
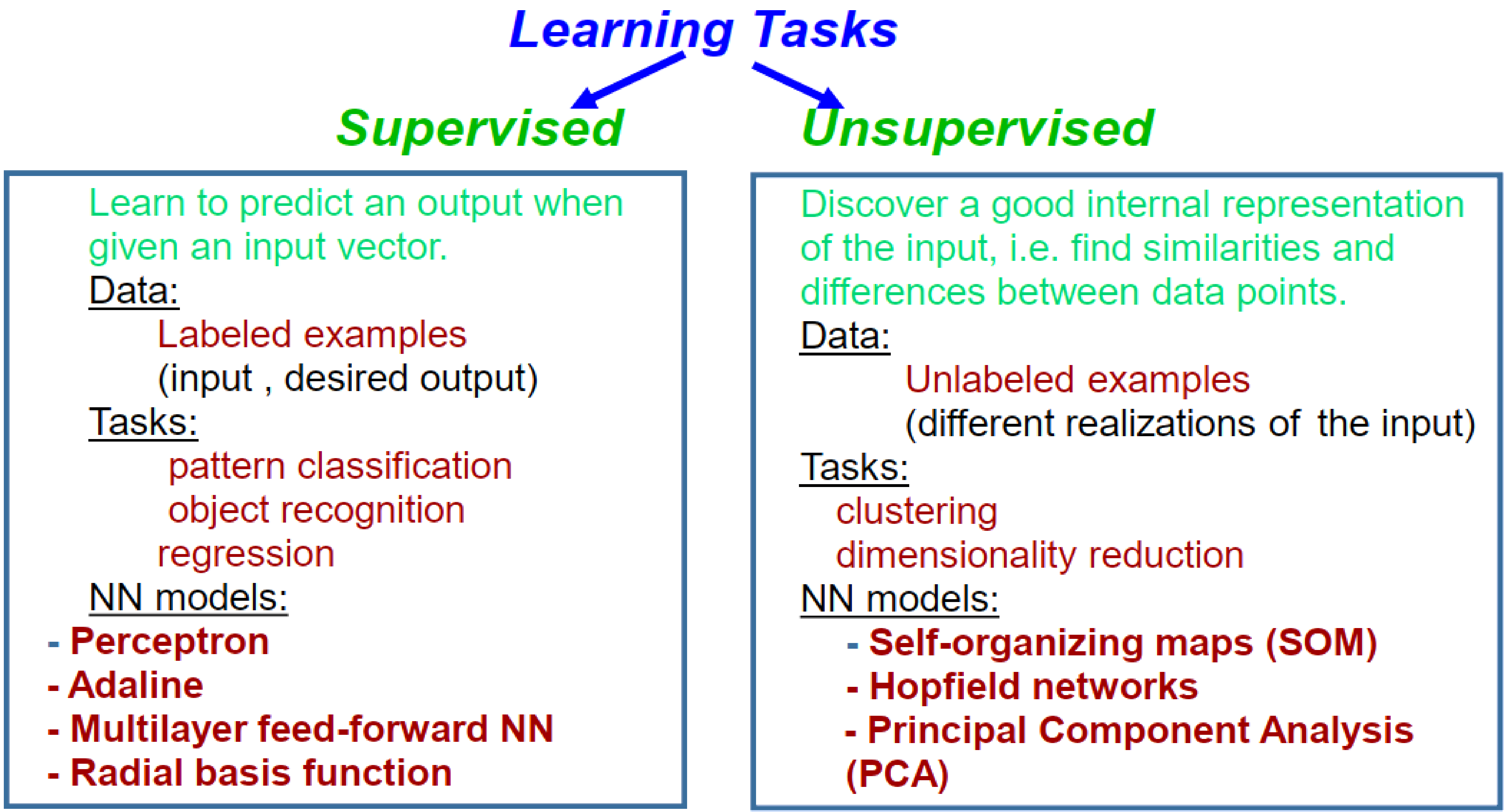

Input pattern x
Unsupervised network for feature extraction
Feature vector y
Supervised network for classification
1
2
r
Outputs
(a)
Feature extraction
Classification
x
y
m-dimensional observation space
q-dimensional feature space
r-dimensional decision space
(b)

Illustration of the classical approach to pattern classification



Learning Tasks

Supervised

Learn to predict an output when given an input vector.

Data:

Labeled examples
(input , desired output)

Tasks:

pattern classification
object recognition
regression

NN models:

- Perceptron
- Adaline
- Multilayer feed-forward NN
- Radial basis function

Unsupervised

Discover a good internal representation of the input, i.e. find similarities and differences between data points.

Data:

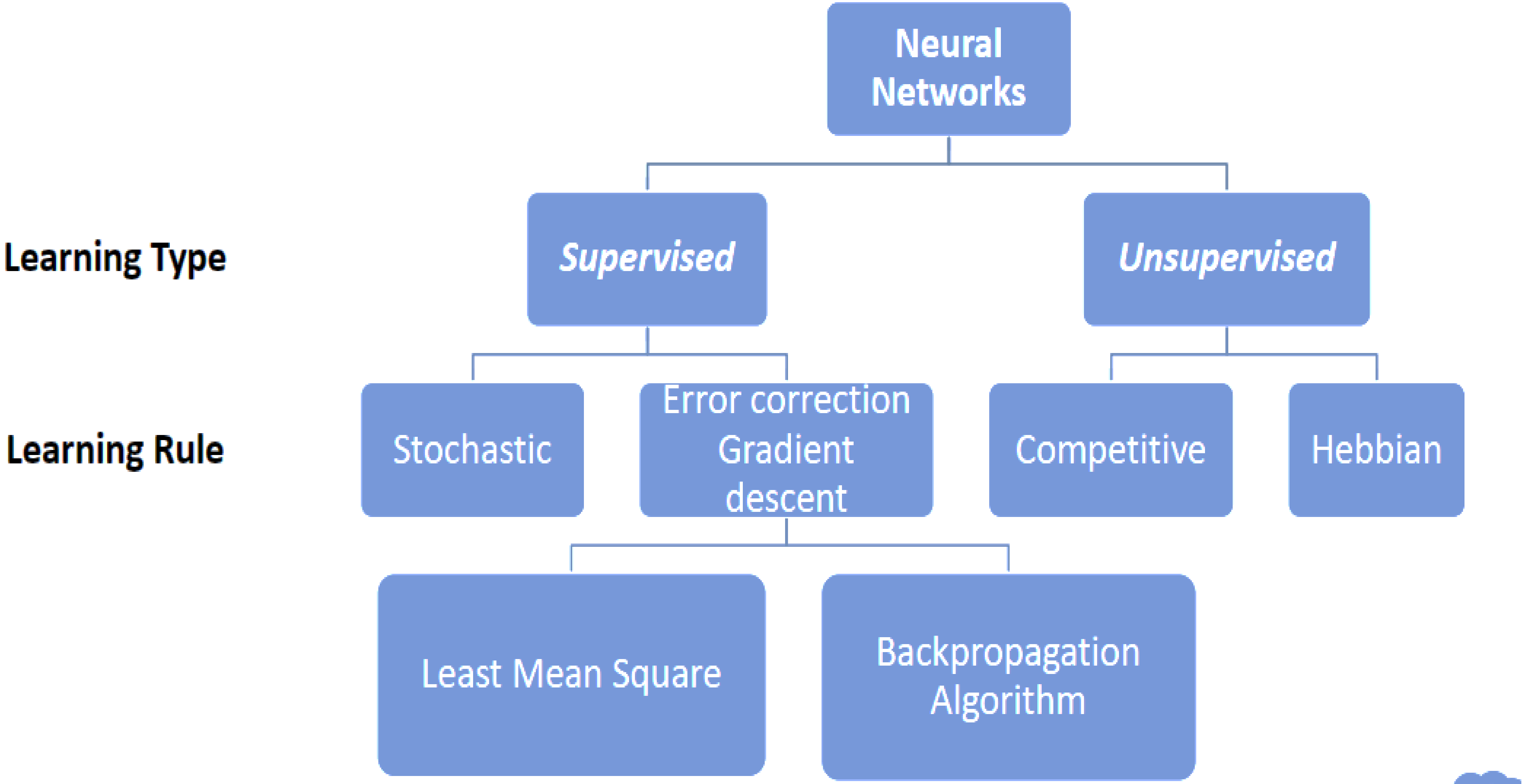
Unlabeled examples
(different realizations of the input)

Tasks:

clustering
dimensionality reduction

NN models:

- Self-organizing maps (SOM)
- Hopfield networks
- Principal Component Analysis (PCA)



Neural Networks

Learning Type

Supervised

Unsupervised

Learning Rule

Stochastic

Error correction Gradient descent

Competitive

Hebbian

Least Mean Square

Backpropagation Algorithm

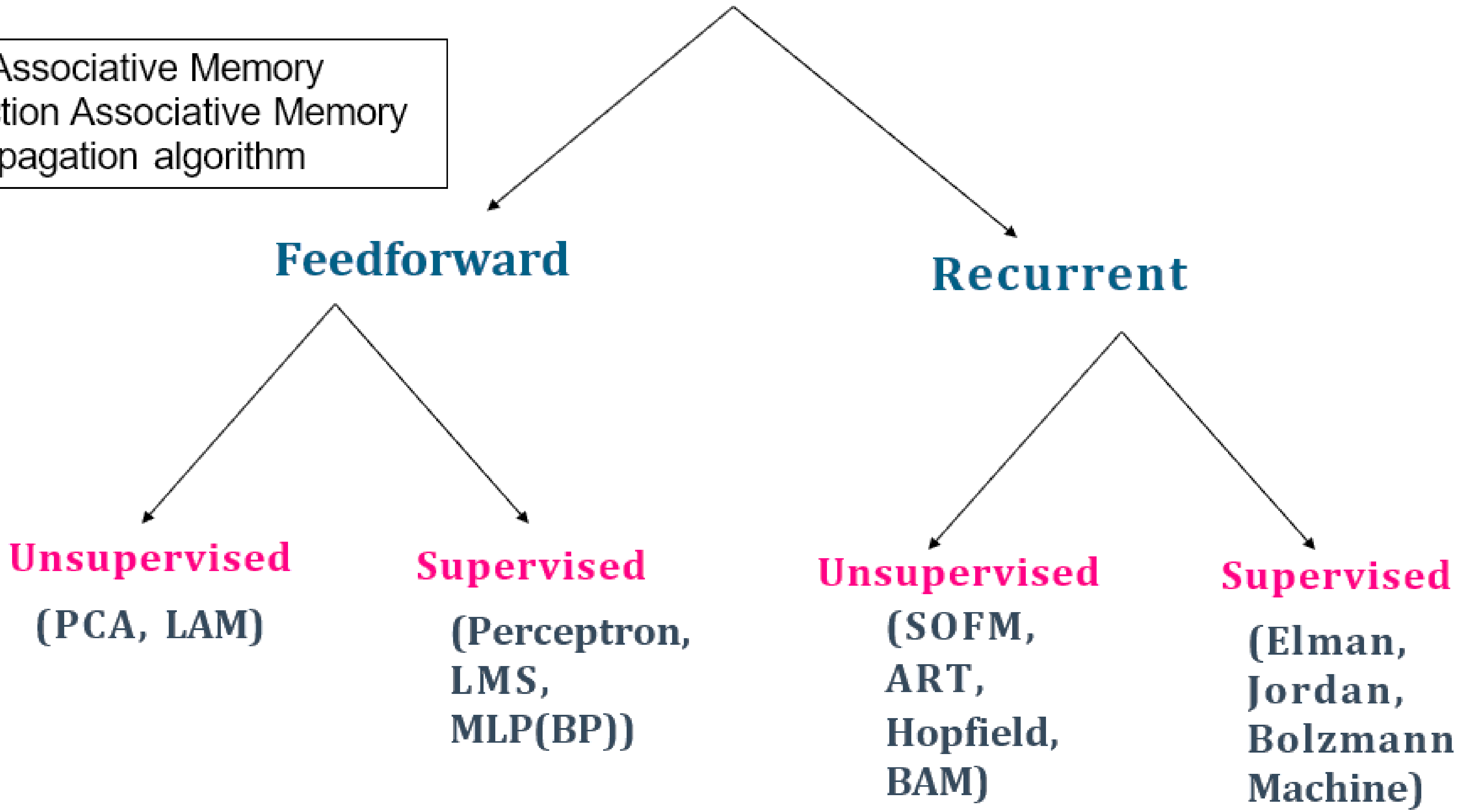
Each of these can be trained with or without a teacher.

Have a particular architecture and learning algorithm.

Taxonomy of neural networks

Based on architecture types and the learning methods

LAM=linear Associative Memory
BAM=bidirection Associative Memory
BP=Backpropagation algorithm



Feedforward

Recurrent

Unsupervised

(PCA, LAM)

Supervised

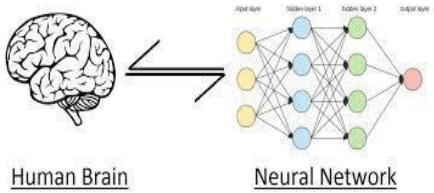
(Perceptron,
LMS,
MLP(BP))

Unsupervised

(SOFM,
ART,
Hopfield,
BAM)

Supervised

(Elman,
Jordan,
Bolzmann
Machine)

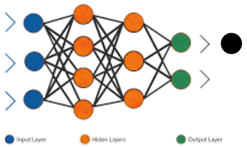


- **Single-layer feedforward networks** and Classification Problems (McCulloch- Pitts neuron model)
  - Decision Surface
  - Rosenblatt’s Perceptron Learning Rule (The Perceptron Convergence Theorem)
  - Multiple-output-Neurons Perceptron

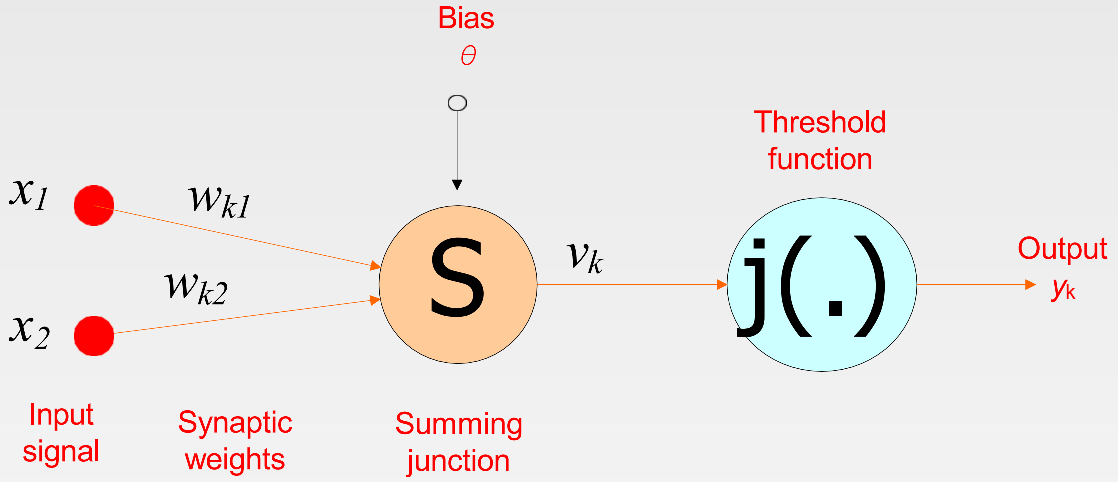
- **Adaline (Adaptive Linear Neuron) Networks**

- **Derivation of the LMS algorithm**

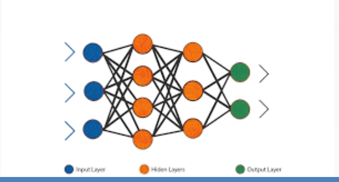
- **Compare ADALINE with Perceptron**



Threshold Logic Units (TLU)



$$y_k = \varphi(v_k) = \varphi(w_{k1}x_1 + w_{k2}x_2 + \theta) = \begin{cases} 1 & \text{if} v_k \geq 0 \\ 0 & \text{if} v_k < 0 \end{cases}$$



TLU try to solve Simple Classification Problem:

The goal of pattern classification is to assign a physical object or event to one of a set of a pre-specified classes (or categories)

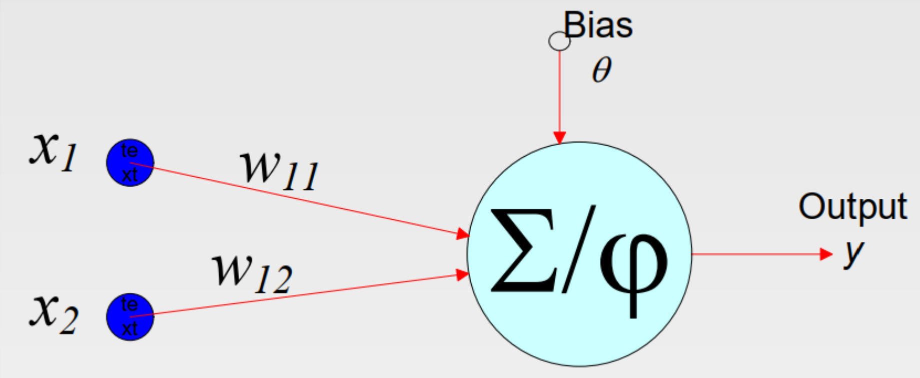
Examples:

- **Boolean functions**
- **Pixel Patterns**

The AND problem

x_1	x_2	y
1	1	1
1	0	0
0	1	0
0	0	0

$Let \quad w_{11} = w_{12} = 1$



$$with \quad x_1 = 1 \quad and \quad x_2 = 1 \quad , \quad y = \varphi(1\times1 + 1\times1 + \theta) = \varphi(2+\theta) = 1 \quad if\,(2+\theta \geq 0)$$

$$with \quad x_1 = 1 \quad and \quad x_2 = 0 \quad , \quad y = \varphi(1\times1 + 1\times0 + \theta) = \varphi(1+\theta) = 0 \quad if\,(1+\theta < 0)$$

$$with \quad x_1 = 0 \quad and \quad x_2 = 1 \quad , \quad y = \varphi(1\times0 + 1\times1 + \theta) = \varphi(1+\theta) = 0 \quad if\,(1+\theta < 0)$$

$$with \quad x_1 = 0 \quad and \quad x_2 = 0 \quad , \quad y = \varphi(1\times0 + 0\times1 + \theta) = \varphi(\theta) = 0 \quad if\,(\theta < 0)$$

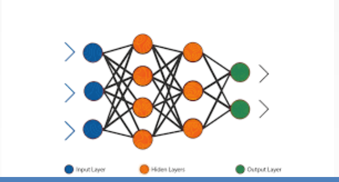
$2 + \theta \geq 0$

$1 + \theta < 0, \quad \theta < 0$

$\theta \quad ?$

$-2 \leq \theta < -1$

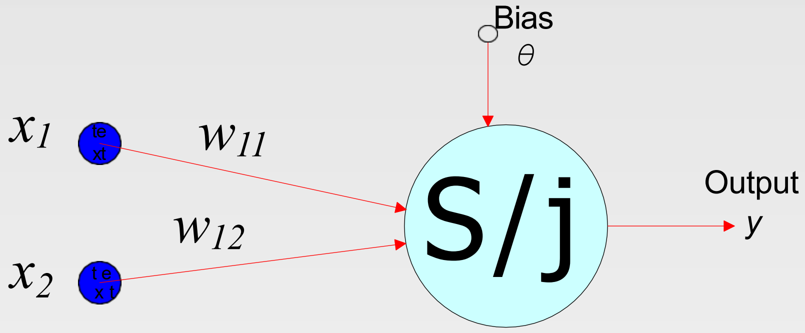
$\theta = -1.5$



The OR problem

x_1	x_2	y
1	1	1
1	0	1
0	1	1
0	0	0

$$Let \quad w_{11} = w_{12} = 1$$



$$with \quad x_1 = 1 \quad and \quad x_2 = 1 \quad , \quad y = \varphi(1\times1+1\times1+\theta) = \varphi(2+\theta) = 1 \quad if\,(2+\theta \geq 0)$$

$$with \quad x_1 = 1 \quad and \quad x_2 = 0 \quad , \quad y = \varphi(1\times1+1\times0+\theta) = \varphi(1+\theta) = 1 \quad if\,(1+\theta \geq 0)$$

$$with \quad x_1 = 0 \quad and \quad x_2 = 1 \quad , \quad y = \varphi(1\times0+1\times1+\theta) = \varphi(1+\theta) = 1 \quad if\,(1+\theta \geq 0)$$

$$with \quad x_1 = 0 \quad and \quad x_2 = 0 \quad , \quad y = \varphi(1\times0+0\times1+\theta) = \varphi(\theta) = 0 \quad if\,(\theta < 0)$$

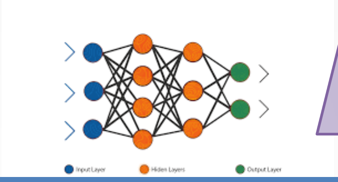
$2+\theta \geq 0$

$1+\theta \geq 0, \quad \theta < 0$

θ ?

$-1 \leq \theta < 0$

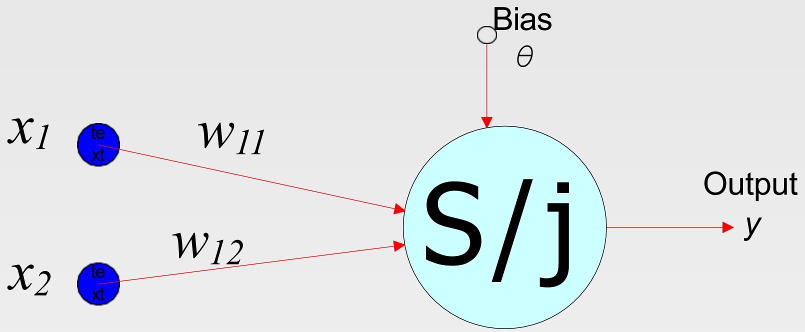
$\theta = -0.5$



The XOR problem

x_1	x_2	y
1	1	0
1	0	1
0	1	1
0	0	0

Let $w_{11} = w_{12} = 1$

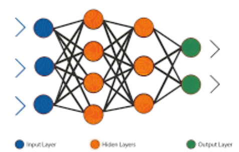


$$\text{with} \quad x_1 = 1 \quad \text{and} \quad x_2 = 1 \quad \text{, the} \quad \text{output} \qquad y = \varphi(1\times1+1\times1+\theta) = \varphi(2+\theta)$$

$$\text{with} \quad x_1 = 1 \quad \text{and} \quad x_2 = 0 \quad \text{, the} \quad \text{output} \qquad y = \varphi(1\times1+1\times0+\theta) = \varphi(1+\theta)$$

$$\text{with} \quad x_1 = 0 \quad \text{and} \quad x_2 = 1 \quad \text{, the} \quad \text{output} \qquad y = \varphi(1\times0+1\times1+\theta) = \varphi(1+\theta)$$

$$\text{with} \quad x_1 = 0 \quad \text{and} \quad x_2 = 0 \quad \text{, the} \quad \text{output} \qquad y = \varphi(1\times0+0\times1+\theta) = \varphi(\theta)$$

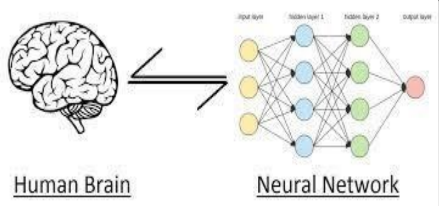


What is the suitable value of θ that make:

1- $2+\theta<0$ and $\theta<0$ to get on an output **y=0**

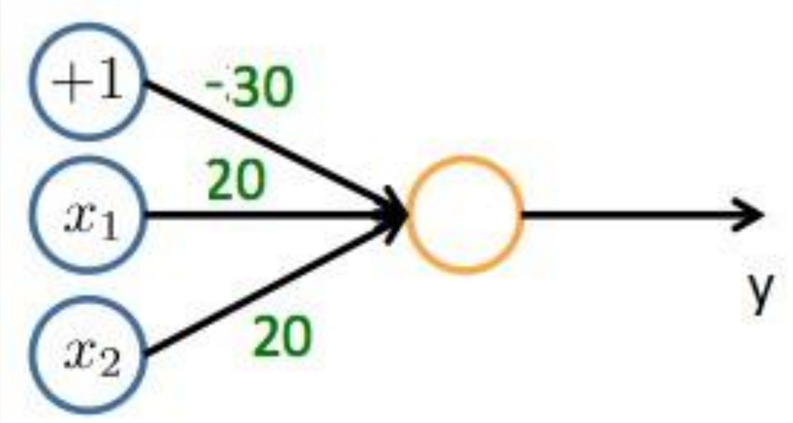
2- $1+\theta>0$ to get on an output **y=1**

No suitable value for θ can solve the XOR problem.



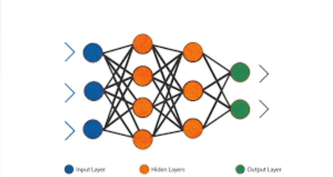
Example

Consider the following neural network which takes two binary-valued inputs x_1, x_2 $\epsilon\{0, 1\}$ and output y. Which one the logical functions does it (approximate) compute?

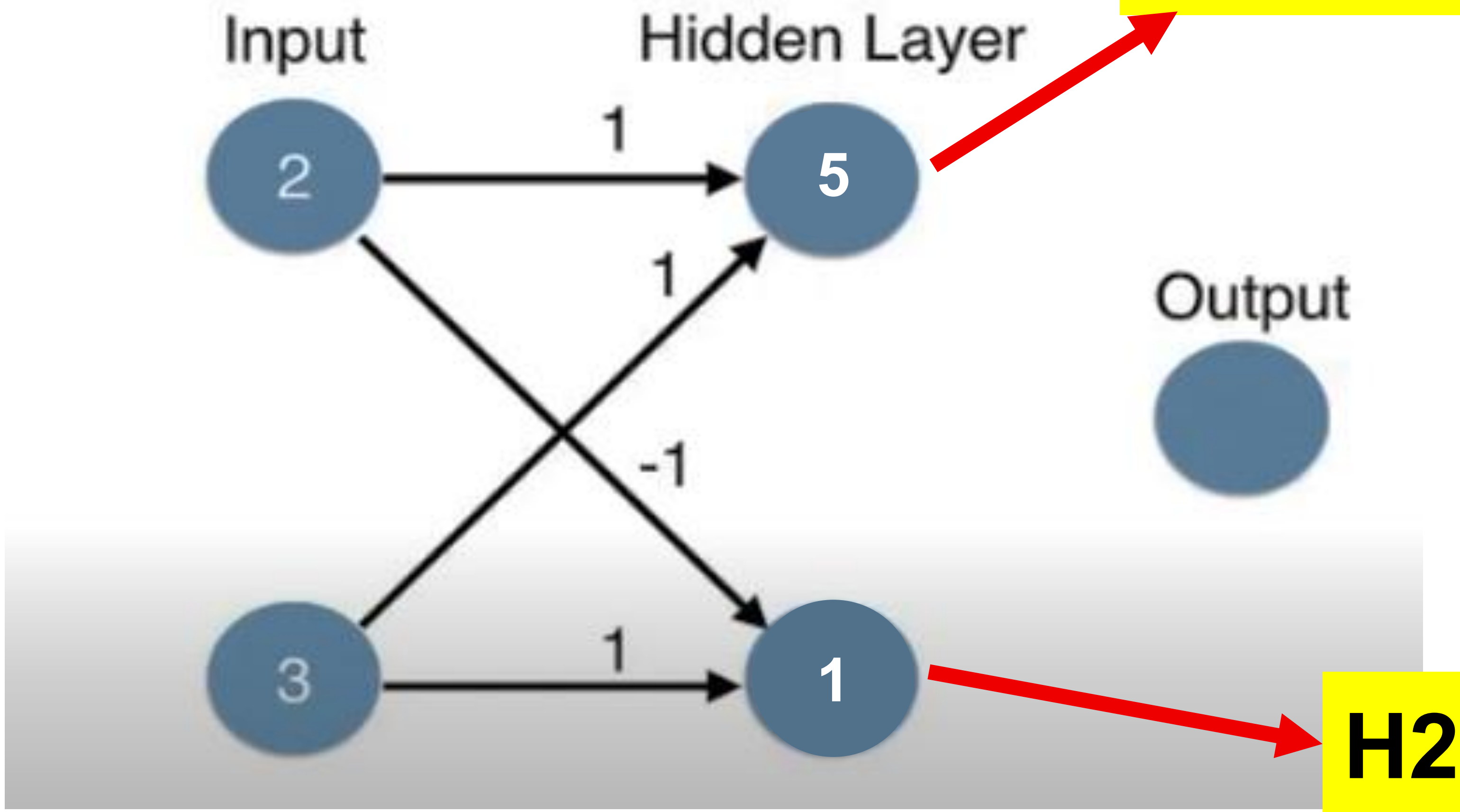


x_1	x_2	y
1	1	
1	0	
0	1	
0	0	

$$y = j(-30 + 20x_1 + 20x_2)$$



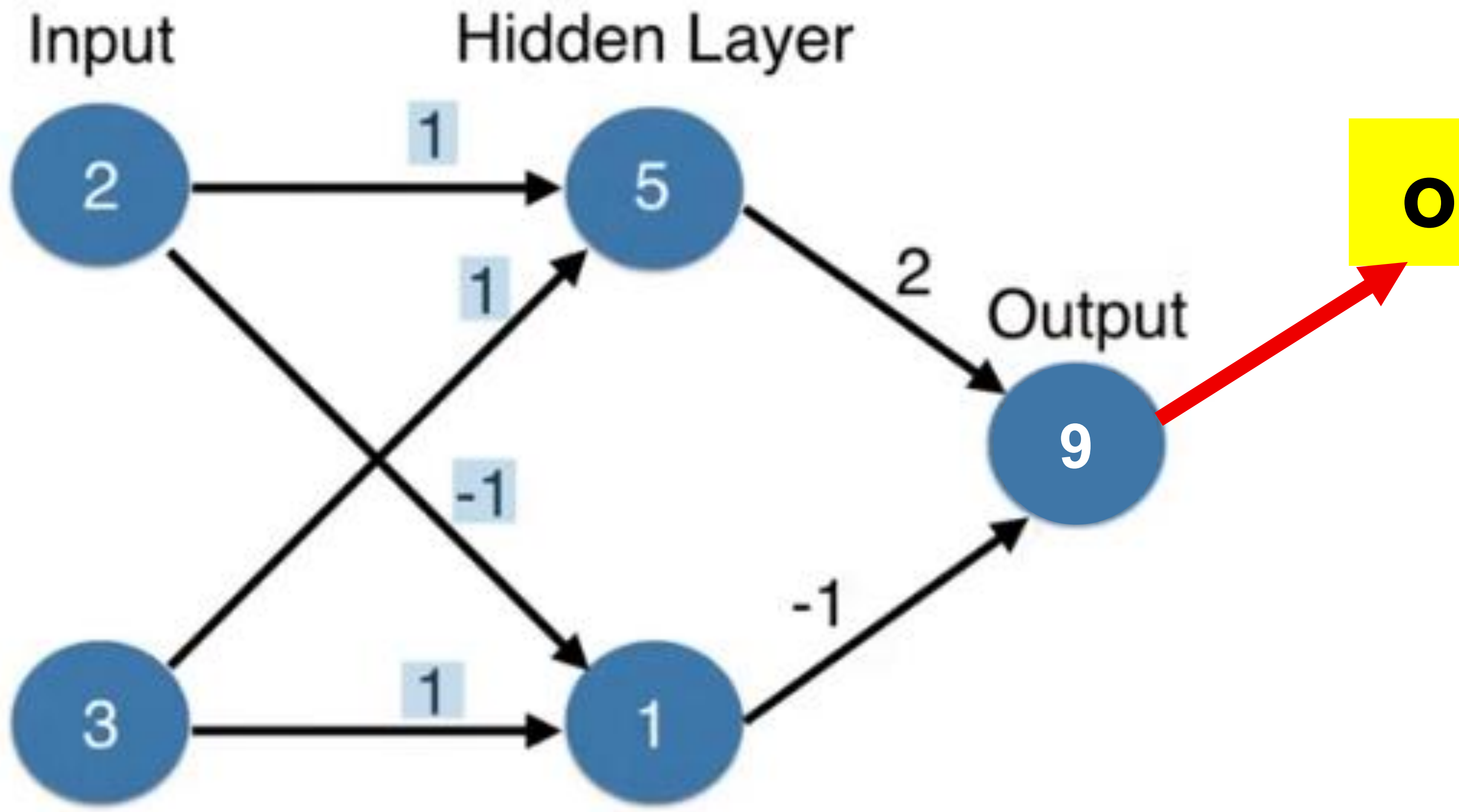
Forward Propagation



H1=2*1+3*1 =5

H2=2*-1+3*1 =1

Forward Propagation



o=5*2+1*-1 =9

Thank You